

1. Environment

- A. OS : Windows 10
- B. compiler version : TDM-GCC 4.9.2 64-bit
- C. IDE : Dev-C++ 5.11

2. Methods or solutions

A. 從 txt 檔讀取陣列

- i. 用 `getline` 逐列讀取並存入一字串，並計算共有幾列，若字串為空，表示該陣列結束。
- ii. 每行讀取完皆將字串以空格為分割標的物，分割並存入 `vector`，最後將 `vector` 的 `size` 除以列數即可得行數。
- iii. 將 `vector` 內的元素逐一轉成 `int` 型別，存入另一動態記憶體配置的整數型別陣列 `arr`。

B. Dynamic program 解決問題

- i. 建一個天數 \times 課程數大小的陣列 `DP`，依序填入合理數值，以範例測資為例：

- 1. 陣列 `arr` 為一陣列，紀錄該課程選幾天會得到多少分數

天數 \ 課程	1	2	3	4
1	3	4	3	6
2	6	6	4	7
3	7	9	8	9
4	8	11	9	10

- 2. 陣列 `DP` 記錄自己與前面所有課程組成某一天數最多可獲得多少分數

還的天數 \ 課程	1	2	3	4
1	3	0	0	0
2	6	7	0	0
3	7	10	10	0
4	8	12	13	16
5	0	15	15	19
6	0	0	18	21
7	0	0	0	24

- A. 將陣列 `DP` 初始化，所有位置皆先填入 0
- B. 陣列 (1, 1) 位置表課程一選一天所獲得的分數，故填入 3
- C. (1, 2)、(1, 3)、(1, 4) 皆為課程 1 分別選 2、3、4 天所獲得的

分數


- D. (1, 5)、(1, 6)、(1, 7)皆填入 0 因課程 1 不能選到 5 天(含)以上，否則不符合個課程皆須至少選一天的規則，可推得當總和天數 > (天數 - 剩下課程數) 為不合法
 - E. (2, 1)也填入 0，因(2, 1)表課程 1、2 共選兩天的情況，但因規定各課程需各選一天，故此格也不合法，因此填入 0
 - F. (2, 2)即課程 1、2 共選 2 天，故必然各選一天，此格的值即為 $DP[1][1] + arr[1][2]$
 - G. (2, 3)為課程 1、2 共選 3 天，故將 $DP[1][1] + arr[2][2]$ 、 $DP[2][1] + arr[1][2]$ ，取最大值，即為課程 1、2 共選 3 天的最大成績
 - H. 其他位置皆仿造上面的規則，最後(7, 4)即為所有課程組成 7 天所能獲得的最大值
3. 由上述根據範例測資，推得的陣列填法，可得下列解法
- A. 每一格可能為下列三種情況的其中一種
 - i. 總共選的天數小於前面所有課程數及自己，違反各選一天的規則，故填入 0
 - ii. 總共選的天數等於前面所有課程數及自己，若該格為 $DP[i][j]$ ，則此格填入 $DP[i-1][j-1] + arr[1][j]$
 - iii. 總共選的天數(i)大於前面所有課程數及自己(j)，若該格為 $DP[i][j]$ ， $DP[i][j] = \max(DP[i-k][j-1] + arr[k][j], DP[i][j])$ ，其中 $1 \leq k \leq i - (j-1)$ (即至少選一天，至多選到目前總共選的天數 - 前面的課程數 (不含自己) 且不可超過總天數 - 所有課程數 + 1)

3. Results

A. Dev C++ 編譯並執行後的結果

```
C:\Users\user\Desktop\演算法\作業\Dynamic Program\DP.exe
24
19
18
13
-----
Process exited after 0.06878 seconds with return value 0
請按任意鍵繼續 . . .
```

B. Output.txt

 output - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

24

19

18

13|